

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method of autonomically patching computer program code, comprising the steps of:
 - executing a computer program instruction, wherein the computer program instruction is located at [[the]] a start of a block of code of an execution sequence of original code instructions;
 - determining [[if]] whether metadata is associated with the computer program instruction, wherein the metadata identifies the computer program instruction as a computer program instruction having associated patch instructions, and indicates an address of the patch instructions;
 - [[if]] responsive to determining that metadata is associated with the computer program instruction is found, redirecting execution to the patch instructions at the address indicated by the metadata;
 - executing the patch instructions;
 - returning to an instruction of the execution sequence of original code instructions in the computer program; and
 - storing a result of executing the execution sequence.
2. (Original) The method of claim 1, wherein the patch instructions are created during execution of the computer program.
3. (Currently amended) The method of claim 1, wherein the patch instructions are created by:
 - copying instructions from the block of code to a new memory location;
 - modifying the order of the instructions of the block of code; and
 - populating the metadata with a pointer to the patch instructions.
4. (Original) The method of claim 1, wherein the metadata is in a form of a memory word.
5. (Currently amended) The method of claim 1, wherein the metadata includes a pointer to the patch instructions for indicating the address of the patch instructions.
6. (Original) The method of claim 5, wherein the pointer to the patch instructions includes a starting address of the patch instructions in an allocated memory location.

7. (Original) The method of claim 6, wherein the starting address includes at least one of an absolute or offset address.

8. (Currently amended) The method of claim 1, further comprising checking a [[new]] flag in a machine status register to determine whether code patching functionality is to be enabled; and responsive to determining that code patching functionality is to be enabled, enabling the code patching functionality.

9. (Currently amended) The method of claim 1, wherein the patch instructions ~~include~~ includes at least one of reorganized instructions, instrumented alternative instructions, and hooks to build an instruction trace.

[[[11]]] 10. (Currently amended) A data processing system having a processor for autonomically patching computer program code, the data processing system comprising:

executing means for executing a computer program instruction, wherein the computer program instruction is located at [[the]] a start of a block of code of an execution sequence of original code instructions;

determining means for determining [[if]] whether metadata is associated with the computer program instruction;

[[if]] redirecting means, responsive to determining that metadata is associated with the computer program instruction is found, for redirecting execution to the patch instructions at the address indicated by the metadata;

executing means for executing the patch instructions;

returning means for returning to an instruction of the execution sequence of original code instructions in the computer program; and

storing means for storing a result of executing the execution sequence.

[[[12]]] 11. (Currently amended) The data processing system of claim [[11]] 10], wherein the patch instructions are created during execution of the computer program.

[[[13]]] 12. (Currently amended) The data processing system of claim [[11]] 10], wherein the patch instructions are created by:

copying instructions from the block of code to a new memory location;

modifying the order of the instructions of the block of code; and
populating the metadata with a pointer to the patch instructions.

[[14]] 13. (Currently amended) The data processing system of claim [[11]] 10, wherein the metadata is in a form of a memory word.

[[15]] 14. (Currently amended) The data processing system of claim [[11]] 10, wherein the metadata includes a pointer to the patch instructions for indicating the address of the patch instructions.

[[16]] 15. (Currently amended) The data processing system of claim [[15]] 14, wherein the pointer to the patch instructions includes a starting address of the patch instructions in an allocated memory location.

[[17]] 16. (Currently amended) The data processing system of claim [[16]] 15, wherein the starting address includes at least one of an absolute or offset address.

[[18]] 17. (Currently amended) The data processing system of claim [[11]] 10, further comprising checking means for checking a [[new]] flag in a machine status register to determine whether code patching functionality is to be enabled; and
enabling means, responsive to determining that code patching functionality is to be enabled, for enabling the code patching functionality.

[[19]] 18. (Currently amended) The data processing system of claim [[11]] 10, wherein the patch instructions ~~include~~ includes at least one of reorganized instructions, instrumented alternative instructions, and hooks to build an instruction trace.

[[20]] 19. (Currently amended) A computer program produced in a computer readable storage medium for autonomically patching computer program code, the computer program product comprising:
first instructions for executing a computer program instruction, wherein the computer program instruction is located at [[the]] a start of a block of code of an execution sequence of original code instructions;

second instructions for determining [[if]] whether metadata is associated with the computer program instruction;

[[if]] responsive to determining that metadata is associated with the computer program instruction is found, third instructions for redirecting execution to the patch instructions at the address indicated by the metadata;

fourth instructions for executing the patch instructions;

fifth instructions for returning to an instruction of the execution sequence of original code instructions in the computer program; and

sixth instructions for storing a result of executing the execution sequence.

[[21]] 20. (Currently amended) The computer program product of claim [[20]] 19, wherein the patch instructions are created during execution of the computer program.

[[22]] 21. (Currently amended) The computer program product of claim [[20]] 19, wherein the patch instructions are created by copying instructions from the block of code to a new memory location; modifying the order of the instructions of the block of code; and populating the metadata with a pointer to the patch instructions.

[[23]] 22. (Currently amended) The computer program product of claim [[20]] 19, wherein the metadata is in a form of a memory word.

[[24]] 23. (Currently amended) The computer program product of claim [[20]] 19, wherein the metadata includes a pointer to the patch instructions for indicating the address of the patch instructions.

[[25]] 24. (Currently amended) The computer program product of claim [[24]] 23, wherein the pointer to the patch instructions includes a starting address of the patch instructions in an allocated memory location.

[[26]] 25. (Currently amended) The computer program product of claim [[25]] 24, wherein the starting address includes at least one of an absolute or offset address.

[[27]] 26. (Currently amended) The computer program product of claim [[20]] 19, further comprising [[sixth]] seventh instructions for ~~checking means for~~ checking a [[new]] flag in a machine status register to determine whether code patching functionality is to be enabled; and eighth instructions, responsive to determining that code patching functionality is to be enabled, for enabling the code patching functionality.

[[28]] 27. (Currently amended) The computer program product of claim [[20]] 19, wherein the patch instructions ~~include~~ includes at least one of reorganized instructions, instrumented alternative instructions, and hooks to build an instruction trace.